

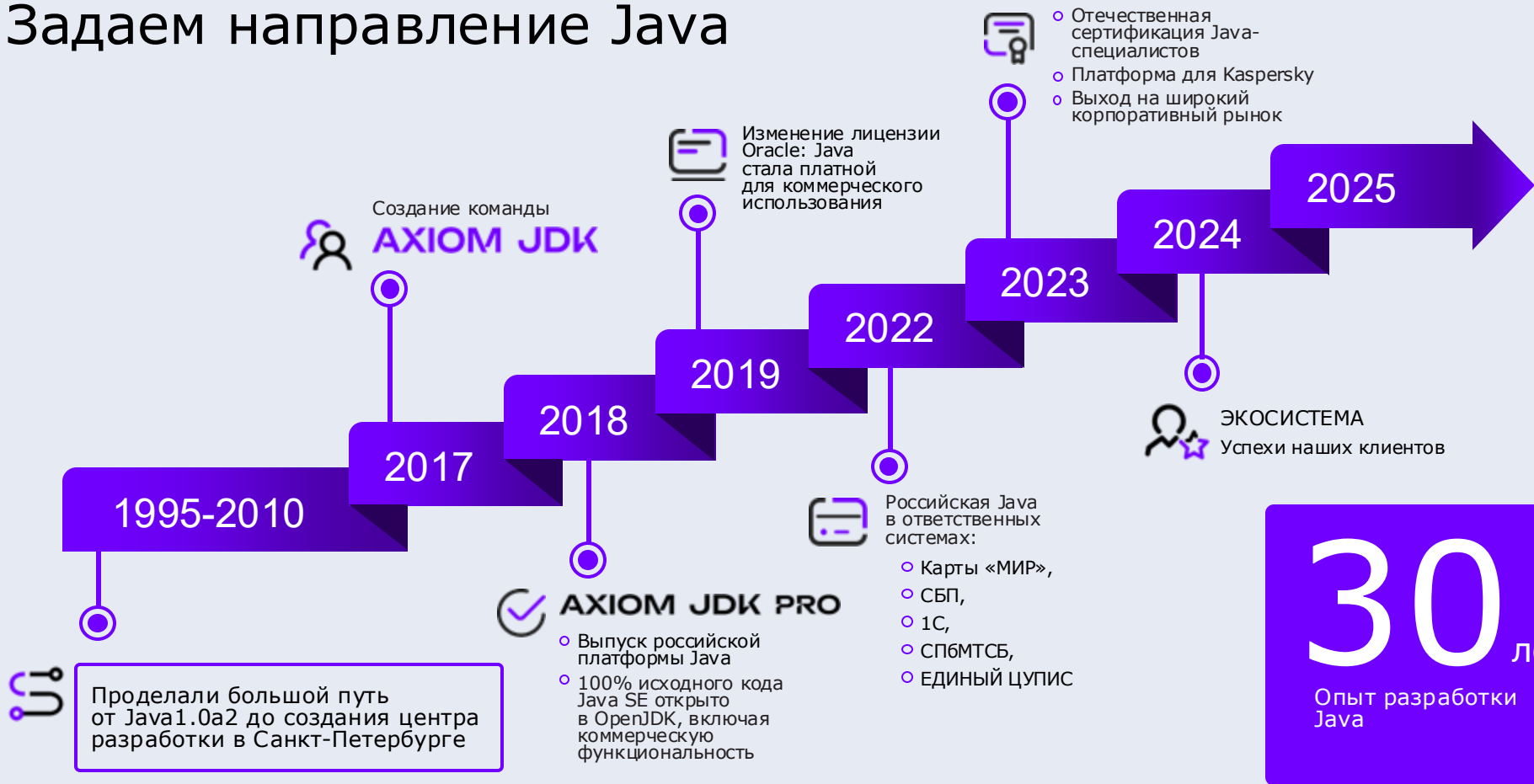
Использование безопасной системы сборки программного обеспечения для продуктов компании Axiom

Алексей Захаров

Директор по технологическому консалтингу Axiom JDK

AXIOM JDK

Задаем направление Java



JAVA – ВАЖНОЕ МЕСТО В ИТ-ЛАНДШАФТЕ КЛЮЧЕВЫХ ОТРАСЛЕЙ РОССИЙСКОЙ ЭКОНОМИКИ

● Защита инвестиций в ПО на Java

**30 000
приложений**

на Java в России

**Каждая 5-ая
вакансия**

в России связана с Java

Примеры отраслевого присутствия:

- **Госорганы > 50%** — Госуслуги, Минцифры
- **Финтех ~ 90%** — ЦБ, МИР, НСПК, ПСБ, ЦФТ, Мобильная карта, СпБМТСБ
- **Нефтегаз ~ 40%** — Газпром, Транснефть
- **Энергетика ~ 20%** — Гринатом, СОЕС

Кодовая база Axiom JDK

- Объем верифицированного исходного кода российского дистрибутива среды исполнения (JRE) составляет

16 млн. строк

- У нас 6 LTS релизов **94 млн. строк**

- Используем тесты для подтверждения соответствия каждому пункту спецификации

JDK – это самый сложный продукт.
всего на исследование кода ушло

100+ инженеро-лет

Продукты реализуют спецификацию, в которой

>1500 страниц,
например, JVM 23

- Релизы JDK день в день с Oracle
- Широкий спектр ОС и архитектур
 - Windows, Mac, Linux (включая российские)
 - X86_64, SPARC, PPC, S390, AArch64 (включая российские)

Кодовая база Libercat

- Объем верифицированного исходного кода российского дистрибутива Libercat составляет **> 400 000 строк**
- У нас 10 релизов включая сертифицированную версию **20 млн. строк**
- Используем тесты для подтверждения соответствия каждому пункту спецификации

Libercat – отличается от Tomcat:

- Поддержка распределенных транзакций
- Центр управления парком серверов (отображение CVE в консоли)
- Мониторинг и управление
- JMS - корректный запуск и останов
- Журналирование

Продукты реализуют спецификации

- Java EE
- Jakarta EE

Идентифицированная проблема:

При особой, очень редкой конфигурации параметров сервера приложений появляется возможность обойти проверку пароля

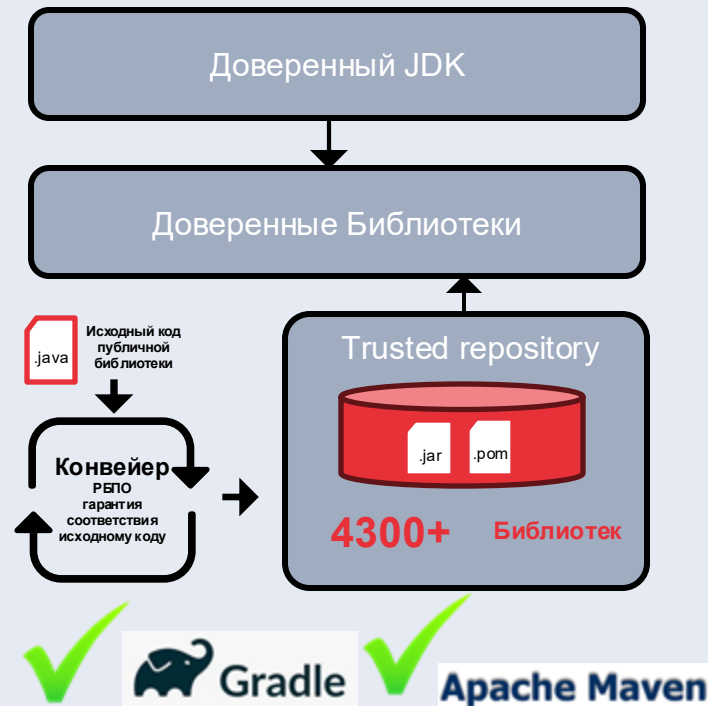
Решение:

Добавили выброс исключения для соответствия спецификации, устранив проблему

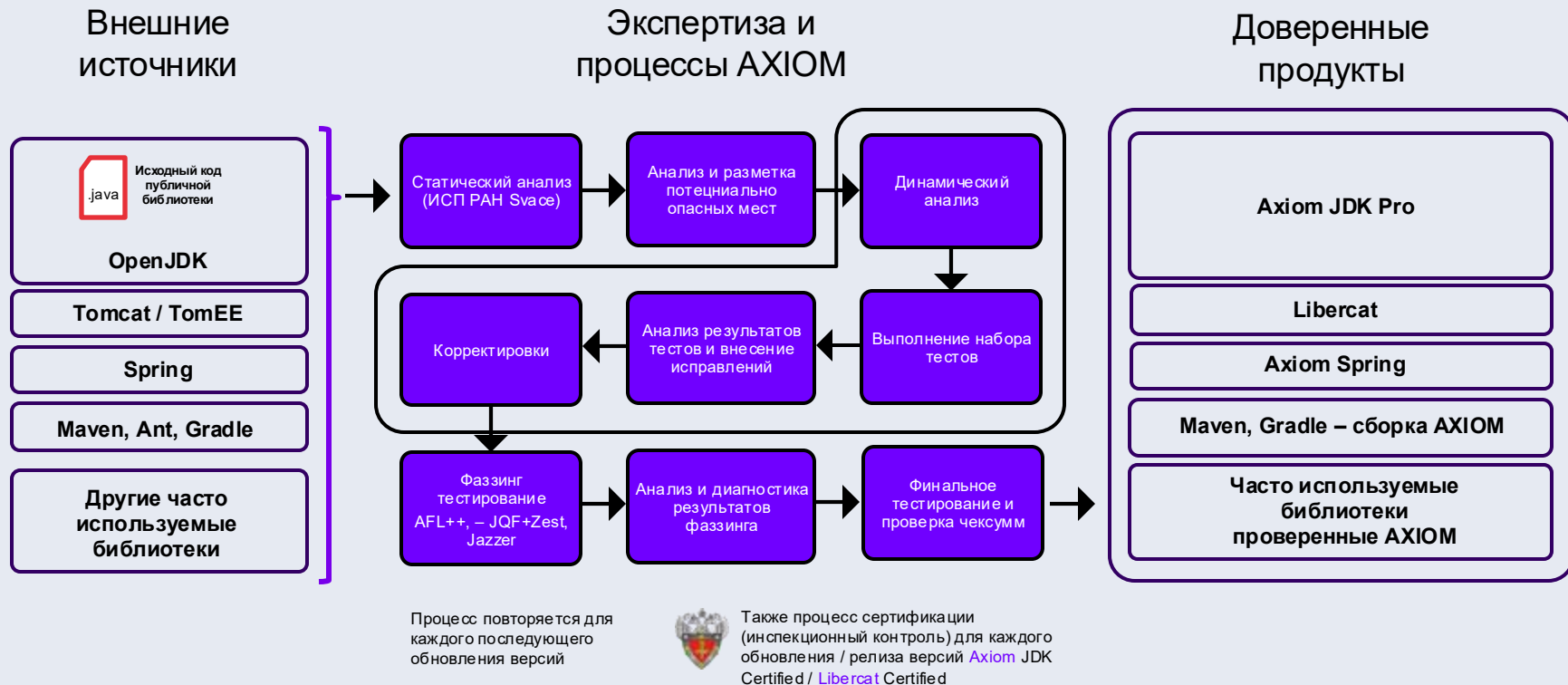
ДОВЕРЕННЫЙ РЕПОЗИТОРИЙ БИБЛИОТЕК

- Средства сборки собираем с помощью нашего безопасного компилятора Java который уже прошел проверку
- Пользуемся ими для компиляции библиотек в Доверенном репозитории
- Разработали доверенный репозиторий Java библиотек
- Сделали конвейер – скорость обработки 25 библиотек в день, стремимся ускорить процесс до 100 библиотек в день
- Планируем добавить SBOM к каждой библиотеке до конца 2025

Добавляем инструменты исследования кода к конвейеру



AXIOM РБПО И ДОВЕРЕННЫЙ РЕПОЗИТОРИЙ



Собираем безопасную систему сборки

- Исходный репозиторий от вендора
- Клонировем репозиторий к себе, собираем проект
- Проводим статический анализ кода при помощи анализатора Svmc (ИСП РАН), PVS-Studio
- Запускаем тесты, Unit, Функциональные тесты, Антивирусы (Kaspersky и DrWeb) для идентификации вредоносного кода
- Верификация - собранный артефакт функционирует, добавляем нашу подпись, кол-во артефактов, версии Java классов, Byte code, проверяем на наличие Shaded Jar
- Делаем комплект файлов для публикации
- Подписываем артефакты (с помощью ключа)
- Создали Java агент поскольку это не просто и Maven переделывает подпись () для добавления "Маркера" в файл манифеста что сборка произведена Axiom JDK



Apache Maven

Сложно менять, нужно менять исходный код, нужно обеспечивать совместимость с плагинами



 **Gradle**

Более простой

Maven и Gradle – Собранны и успешно им пользуемся и они будут доступны в 4 квартале 2025 года

Axiom Repo – Добавление библиотек нелинейный процесс

- Объем верифицированного исходного кода
> 50 000 000 строк
- У нас **4300+** библиотек по 5-20 тысяч строк

Скорость обработки зависит от их сложности и зависимостей, некоторые библиотеки обрабатываются в течение 2х месяцев

Axiom Repo – основные отличия

- Оповещение о CVE по почте и в личном кабинете
- Запрос через API (библиотек с CVE) в процессе работы сборочного конвейера
- Манифест в JAR

```
Manifest-Version: 1.0
Built-By: Axiom JSC <https://axiomjdk.ru>
Build-Jdk: 1.8.0_442 (Axiom JSC 25.442-b07)
```

Идентифицированная проблема:

Библиотека jackson-core, jackson-databind, в версии 2.12.7-2.15.0-rc3 были CVE начиная с 2.15.0 их нет есть еще в **Woodstox** похожая ситуация с 6.2.6 по 6.3.1

Решение:

Оповещение заказчиков письмом и через личный кабинет, планируем запустить API и SBOM до конца 2025 года

Пример библиотеки Spring-data-commons

- Spring Data Commons — подпроект проекта Spring Data, который предоставляет общую инфраструктуру для модулей Spring Data, связанных с источниками данных.
- Цель — унифицировать доступ к данным, поддерживая различные хранилища, включая реляционные базы данных, NoSQL-хранилища и другие.
- Одна из самых популярных библиотек фрейворка

- Успешный тест
- Исходник
- Обработка
- Результат в репозитории Axiom JDK

```
[2025-09-22T15:02:02.787Z] Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.
[2025-09-22T15:02:02.788Z]     Affected argument(s) used the following variable(s): [SSH_IDENTITY]
[2025-09-22T15:02:02.788Z]     See https://jenkins.io/redirect/groovy-string-interpolation for details.
[2025-09-22T15:02:03.378Z] + install -m600 /dev/null /home/axiom/.ssh/id_rsa
[2025-09-22T15:02:03.378Z] + cp **** /home/axiom/.ssh/id_rsa
[2025-09-22T15:02:03.378Z] + sed -e 's/\.git$//' -e s:trusted-repo/buildspecs:::
[2025-09-22T15:02:03.378Z] + echo trusted-repo/buildspecs/github.com/spring-projects/spring-data-commons.git
[2025-09-22T15:02:03.378Z] + kosa project deliver github.com/spring-projects/spring-data-commons refs/tags/3.0.2
[2025-09-22T15:02:05.915Z] projectPath: github.com/spring-projects/spring-data-commons
[2025-09-22T15:02:05.915Z] releaseTag: refs/tags/3.0.2
[2025-09-22T15:02:10.107Z] MergeRequest[id=11384, iid=36, title=Добавить новую версию проекта 'https://github.com/spring-projects/spring-data-commons.git' по тэгу '3.0.2', description=
[2025-09-22T15:02:10.107Z]   - [Описание задачи](#37)
[2025-09-22T15:02:10.107Z]   - [Репозиторий исходного кода](https://gitlab.int.axiomjdr.ru/trusted-repo/sources/github.com/spring-projects/spring-data-commons), sourceBranch=spring-data-commons-3.0.2, targetBranch=3.0.x, labels=[workflow:Awaiting Review, status::Ready, request::New Version], squash=true, assigneeId=0, forceRemoveSourceBranch=true, state=merged, squashCommitSha=fed9fa16fb82c024685a35ddd4d9a5aaa29c6a96]
```

Упавший тест - надо разбираться

- Пример ошибки сборки драйвера pgjdbc-42.0.0
- Данная ошибка требует исследования почему Maven не смог провести тест
- Данная ошибка привел к невозможности сборки проекта

```
[2025-09-19T14:24:17.095Z] [INFO] -----
[2025-09-19T14:24:17.095Z] [INFO] BUILD FAILURE
[2025-09-19T14:24:17.095Z] [INFO] -----
[2025-09-19T14:24:17.095Z] [INFO] Total time: 08:48 min
[2025-09-19T14:24:17.095Z] [INFO] Finished at: 2025-09-19T14:24:10Z
[2025-09-19T14:24:17.095Z] [INFO] -----
[2025-09-19T14:24:17.095Z] [ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:2.18.1:test (default-test) on project postgresql: There are test failures.
[2025-09-19T14:24:17.095Z] [ERROR]
[2025-09-19T14:24:17.095Z] [ERROR] Please refer to /srv/ws/workspace/trusted-repo/github.com/pgjdbc/pgjdbc/dev-builder/src/pgjdbc-42.0.0/pgjdbc/target/surefire-reports for the individual test
results.
[2025-09-19T14:24:17.095Z] [ERROR] -> [Help 1]
[2025-09-19T14:24:17.095Z] [ERROR]
[2025-09-19T14:24:17.095Z] [ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[2025-09-19T14:24:17.095Z] [ERROR] Re-run Maven using the -X switch to enable full debug logging.
[2025-09-19T14:24:17.095Z] [ERROR]
[2025-09-19T14:24:17.096Z] [ERROR] For more information about the errors and possible solutions, please read the following articles:
[2025-09-19T14:24:17.096Z] [ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojofailureException
[2025-09-19T14:24:17.096Z] + die 'check failed'
[2025-09-19T14:24:17.096Z] + trap - EXIT
[2025-09-19T14:24:17.096Z] + error 'check failed'
[2025-09-19T14:24:17.096Z] + local 'prompt-native-build:'
[2025-09-19T14:24:17.096Z] + printf 'native-build: %s\n' 'check failed'
[2025-09-19T14:24:17.096Z] native-build: check failed
[2025-09-19T14:24:17.096Z] + error_cleanup
[2025-09-19T14:24:17.096Z] + :
[2025-09-19T14:24:17.096Z] + exit 1
script returned exit code 1
```

- Инженеры проводят
детальный анализ
- Устранение ошибок
сборки

Реализация стандарта

5.12 Использование безопасной системы сборки программного обеспечения

5.12.1 Цели

- 5.12.1.1 Обеспечение безопасности при сборке ПО, недопущение привнесения в код ошибок, обусловленных небезопасными преобразованиями кода → Средства статического анализа и антивирусы
- 5.12.2 Требования к реализации
 - 5.12.2.1 Разработать регламент использования системы безопасной сборки ПО → Документирование взаимодействия организовано в Gitlab

Реализация стандарта Описание и Регламенты

5.12.2.2 Для разрабатываемого ПО должна быть зафиксирована информация о системе сборки ПО и сборочной среде.

- Перед сборкой мы составляем мета-описание проекта (buildspec)
- Структурированный набор данных который описывает откуда и что мы берем (Какой Maven, Gradle, Gitlab)
- На основе файла параметров инфраструктуры собираем проект с нужными версиями.

5.12.3.1 Регламент использования системы безопасной сборки ПО должен содержать, как минимум, следующие сведения:

обязанности сотрудников и их роли при выполнении сборки ПО	Назначается ответственный разработчик, после сборки проводится автоматизированное тестирование и ручная проверка группой тестирования
критерии выбора инструментов сборки ПО;	В соответствии с требованиями вендора проекта “Мета описанием”
критерии приемки результатов сборки;	Определено участие команды тестирования и контроля результатов
порядок регистрации событий, генерируемых инструментами сборки ПО	Сохраняем журналы сборки в Gitlab, доступны статусы результатов сборки

Реализация стандарта Информация и Артефакты

5.12.3.2 Информация о сборочной среде должна содержать:

- описание особенностей функционирования сборочной среды; Jenkins + Gitalab + IT infra + библиотеки сборочных инструментов + инструменты (антивирусы, анализаторы (код и сверка),)
- перечень программных инструментов, применяемых в системе сборки ПО, их версий и конфигураций.
- библиотека готовых инструментов расположена в GitLab

5.12.3.3 Артефакты реализации требований, подтверждающие соответствие инструмента из состава системы сборки ПО рекомендациям производителя по безопасному использованию, должны содержать перечень выполненных рекомендаций производителя инструмента сборки ПО с указанием конкретных параметров настроек и конфигураций.

- Реализовано за счет сверки артефактов, с публичными библиотеками
- Журнал сборки который содержит статусы

Автоматизация выпуска релизов

- **Классический пример:** сборка происходит из исходников с помощью `./configure && make`
Результат выгружается в публичные ресурсы, нет гарантии соответствия исходному коду
- **Профессиональный подход - отказоустойчивая инфраструктура для выпуска продукта и обновлений**
Требуется комплект оборудования и ПО, с достаточным набором аппаратных ресурсов

● **Исходники:**



● **Сборки:**



>500
НОД



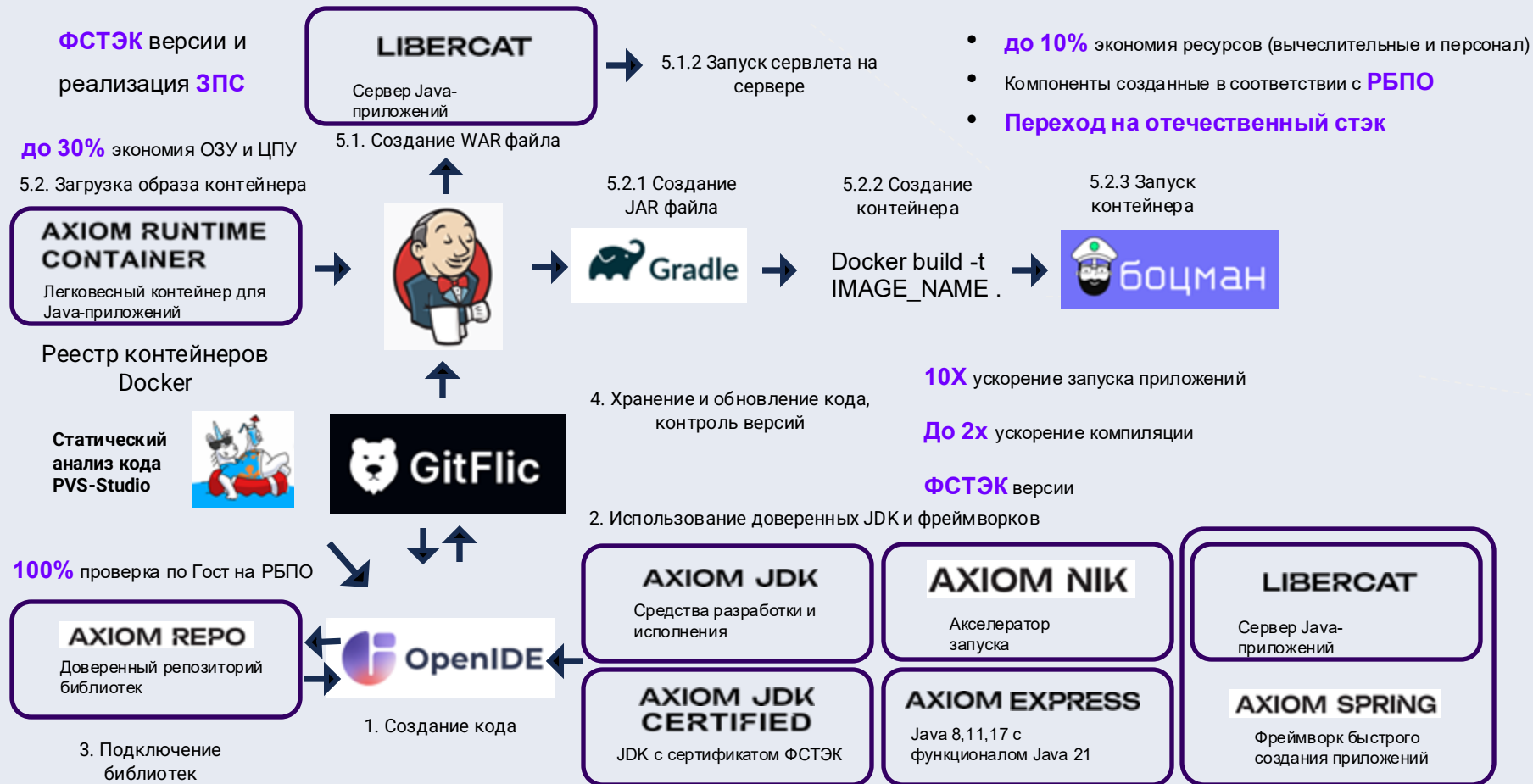
● **Тесты:**

- Статический анализ
- Антивирус
- Функциональные
- Регрессионные
- Производительность

>200
конфигураций

● **Репозитории:**

- Linux
- Container Registry
- Кабинет поддержки
- Axiom Repo
- REST API



ПРЕИМУЩЕСТВА ОТ РЕАЛИЗАЦИИ ПРОЦЕССА БЕЗОПАСНОЙ РАЗРАБОТКИ

- Экономия аппаратных ресурсов **до 15%**
- Экономия ресурсов разработчиков **до 50%**
- Сокращение сроков вывода продуктов на рынок
- Рост производительности пользователей приложений **до 5%**
- Обеспечение безопасности приложений в соответствии с РБПО, не увеличивая нагрузку на подразделение разработки
- Продукты из реестра Минцифры
- Соответствие регуляторике
- Сертификаты ФСТЭК

НАШИ КЛИЕНТЫ И ПАРТНЕРЫ





AXIOM JDK

Стоим на страже безопасности Java



 @axiomjdkpro